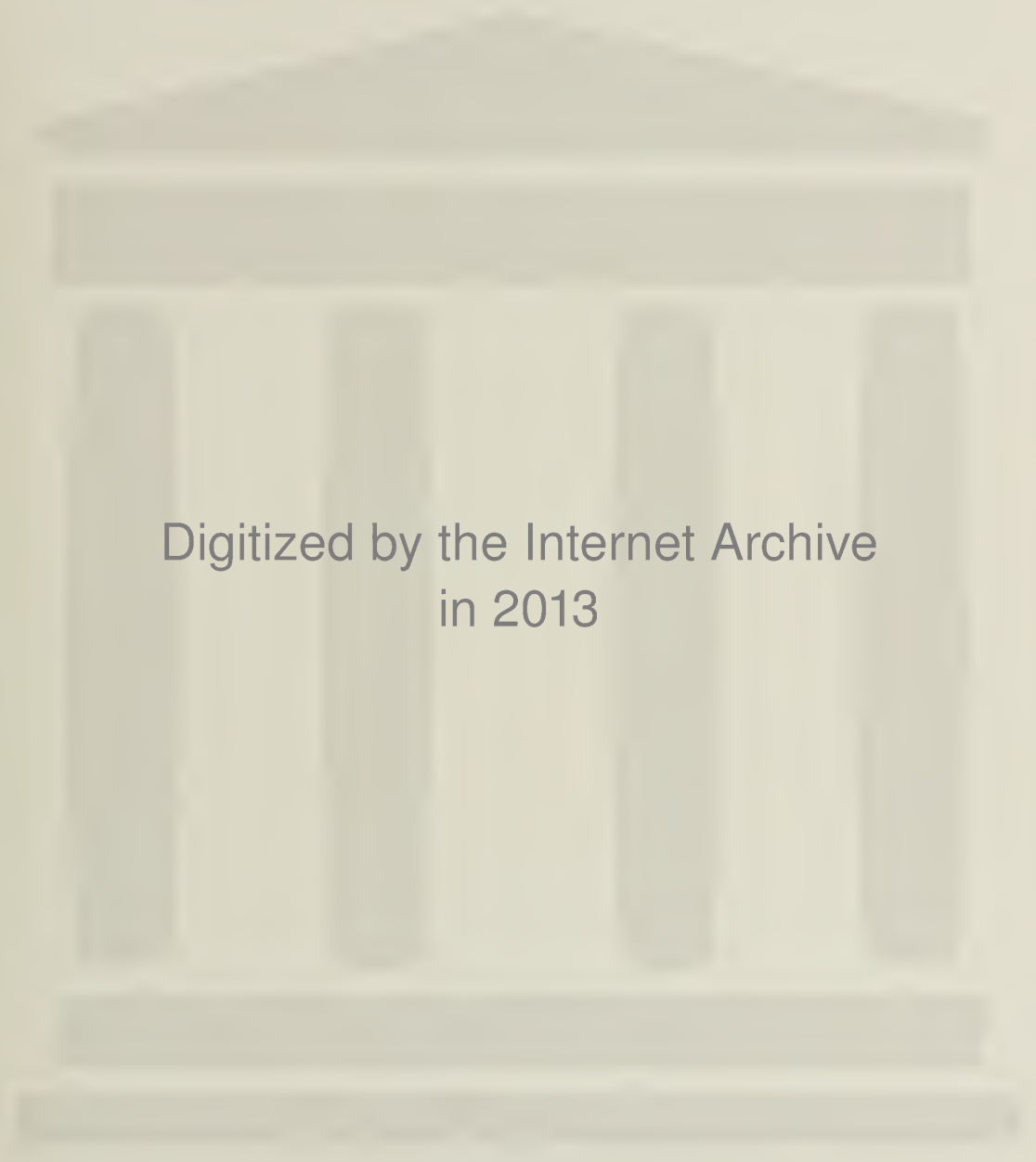

DIGITAL COMPUTER LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

REPORT NO. 151

PROGRAM DESCRIPTION OF PAX
AN IBM 7090 PROGRAM TO SIMULATE THE PATTERN ARTICULATION UNIT OF ILLIAC III

by

James H. Stein

September 16, 1963

This work was supported in part by
the Atomic Energy Commission under
contract No. AT(11-1)-1018.

TABLE OF CONTENTS

1. INTRODUCTION

This report is supplemental to this Laboratory's Report No. 147, "User's Manual for PAX,"* by the author, which presents a "black-box" description of the PAX program. Ambitious programmers may wish to modify or augment that program, and it is to those people that this description of the black box's internal features is addressed.

The PAX program is little more than a collection of closed (or nearly-closed) subroutines together with a simple routine which transfers control to the subroutines in a sequence specified by a list of control words generated by the PAX instructions which are to be simulated.

2. FUNCTIONAL SUBROUTINES

For every operation of the PAU discussed in Report No. 147 there corresponds a functional subroutine in PAX which simulates that operation. All functional subroutines have the following characteristics:

1. Parameters required for the subroutine's operation will be stored in a block of from one to twenty-one 7094 words. The length and format of the parameter-block will vary from subroutine to subroutine; however, the address field of the first word will invariably contain the location of the subroutine which corresponds to the block.

2. At the time of entry to the subroutine, XR7 will contain the two's complement of the address of the first word of the parameter block.

3. Sometime during the operation of the subroutine, XR7 will be decremented by n, where n is the number of 7094 words in the parameter block. (An exception to this is the subroutine which simulates the JUMP orders within PAX.) Thus, when an exit it made from a functional subroutine, XR7 contains the two's complement of the address of the first word of the following parameter block.

---

* Hereafter referred to simply as "Report No. 147."

4. Exits from functional subroutines are usually made by a "TRA* 0,7"* order, i.e., a transfer which is indirectly addressed by XR7. Thus, the exit from a functional subroutine is made (usually) to the functional subroutine which is to process the next parameter block.

3. PARAMETER BLOCKS

The method whereby functional subroutines transfer control to one another is described in Section 2 above. Clearly, the sequencing of the subroutines is dependent upon the sequence of parameter blocks, and this is directly determined by the ordering of the PAX instructions punched on the input cards of the Order Deck (see Appendix I.1 and I.2, Report No. 147).

Each PAX instruction is actually a SCATRE macro call** which expands into a block of from one to twenty-one 7094 words, i.e., a parameter block for the functional subroutine which will simulate the given PAX instruction. For example, the instruction

```
CPRINT    5,36,18,M1,3,M10,6,M20,6,M30,6,M40,6
```

will expand into the five words

```
PZE     PRINT,,36
PZE     ,5,18
SIX     M20,6,M10
SIX     M40,6,M30
PZE     M1,3
```

---

* Actually, control is transferred to a fixed location in the program with the symbolic name "PAX," and it is this location which contains the "TRA* 0,7" order. This method is employed since it allows simple trace routines to be employed (see Appendix III for a sample) when code checking a PAU program.

**See: "SCATRE," University of Illinois Library Routine LI-UØI-SCRE-19-BX, Part II, Section 10, for a description of assembly macros.

where PRINT is the address of the functional subroutine which simulates the CPRINT operation.

The format of the parameter block for each functional subroutine is given in Appendix I, which lists the macro definitions corresponding to the PAX orders.

4. INDEXING

Indexing is modification of instruction arguments at execution time, dependent upon the current values of the index registers. Functional subroutines accomplish indexing by using the closed subroutines (not to be confused with functional subroutines) MØRDER and MØDIFF.

If the parameters in a parameter block are indexed, then the functional routine uses MØRDER to make a copy of the parameter block in another part of memory (at symbolic location ØRDERM). It then enters MØDIFY one or more times, causing the parameters in the copy of the parameter block to be altered according to the present contents of the indicated index registers. The functional subroutine then operates on the modified copy of the parameter block, rather than on the original.

5. MACRO PAU INSTRUCTIONS

Most of the functional subroutines are written entirely in 7090 coding. A few, however, contain both 7090 and PAU instructions. The PAX instruction counter, which usually points to a location in the list of parameter blocks supplied by the user, is temporarily diverted to a list stored as a part of the functional subroutine.

Operations which behaved in this fashion are MARK, TMARK, CØNNEC, TCHAIN, and BØØFUN, and are called macro instructions. The functional subroutines for these instructions, after planting various constants according to the parameters, fake a LINK JUMP to PAU program within the PAX code itself. When a RETURN JUMP is encountered, control returns to the user's PAU program.

The internal PAU code which corresponds to a macro instruction normally uses several planes and index registers. However, the macro instructions behave,

from the user's viewpoint, exactly as normal instructions behave. This is true because the macro instructions change the contents of index registers IR21-25 only (registers which the user does not use), and use for temporary storage an eleven-plane memory which the user also does not use (it is not even mentioned in Report No. 147).

6. SYMBOLIC NAMES

Appendix II of this report gives the listing of the PAX program. A few of the symbolic names used therein are still defined in the condensed, binary PAX deck. (These names are given in Appendix II of Report No. 147.)

Additions or corrections to PAX in SCAT language are easily incorporated into the Order Deck (see Appendix I.1, Report No. 147), since these additions may refer to symbolic locations within the PAX program.

For an example of an addition to the PAX program, see Appendix III.

The significance of the following symbolic names may be of interest to the user:

AND, ØR, EXØR, CLEAR, EQUAL

Entry points to the routines which perform the BØØLØP operation. If "BAR" appears as the eighth argument of the BØØLØP order, then entry is made two locations in front of the normal entry (i.e., to "TYPE-2").

SET, INCR, DECR, TRANS, DSPLY

Entry points to the routines which perform the INDEX operations.

UC, NULL, NØNULL, ZERØ, NØZERO, LESS, EXACT, MØRE, LINK, RETURN

Entry points to the routine which performs the JUMP operation.

MØDIFY, MØRDER

Entry points to the indexing subroutines (see Section 4).

FXMØDE

A location which contains the integer M.

MØDSEL

A location which contains M if M = 36, or 2M if M = 72 (i.e., contains the number of 7094 words necessary to store a plane).

ØRDERM

The first words of a block of 21 words in which copies of parameter blocks are stored during the indexing process (see Section 4).

ØRDERS

A block of 14,600 words into which the PAX user causes PAX programs and/or 7094 programs to be stored, according to the contents of the Order Deck (see Appendix I.1, Report No. 147).

PAX

This location contains the 7094 instruction "TRA 0,7". Most functional subroutines exit to this location (see Section 2).

DL1, DL2, DL37, DLSUR, etc.

Names of direction lists.

M0, M1, ..., M50

Mi is the location of a block of 144 words in which the ith plane of memory is stored.

IR

The first word of a block of 25 words which simulate the contents of IR1-IR25. The contents of IRk are in location IR - 1 + k.

7. ADDITIONAL PAU INSTRUCTIONS

There exist functional subroutines for four PAU orders which are not mentioned in Report No. 147: SETTRA, SETTRC, SETTRB, and SETSET. These operations were defined during an early varsion of PAX, and have been deleted from the order code since their utility is more than replaced by the orders SHIFTD and DLTØIR.

However, the internal PAU programs which correspond to the macro-instructions (see Section 5) include these four old orders. Thus, these four are defined below for completeness:

SETTRA d,NAME

Plant x,y values in the SHIFT order at location NAME so that execution of the SHIFT order will cause a shift of one step in the d direction.

SETTRC d,NAME

Same as SETTRA, except the shift will occur in the -d direction.

SETTRB k,NAME

The x,y values of the SHIFT order at location NAME are multiplied by V(IRk). (Restriction: V(IRk) must be non-negative.)

SETSET DL*,NAME1,NAME2,NAME3

This is a tricky order which is similar in many respects to the DLTØIR instruction.

On the first execution of SETSET, the first direction in the direction list DL is planted in the SETTRA (or SETTRC) order at location NAME1, and the negative of the first direction of the direction list DL is planted in the SETTRA (or SETTRC) order at location NAME2.

Similarly, the ith execution of SETSET results in the ith direction number of the direction list DL being planted at NAME1 and NAME2. Unlike DLTØIR, however, the direction list DL is not necessarily the same direction list at each execution of SETSET, because (unlike DLTØIR) the index register which modified DL (if any) is consulted during every execution of SETSET, not just the first.

This process must terminate, however, for there will be some execution of SETSET (say, the mth) for which the direction list DL contains fewer than m directions. When this occurs, the SETSET order will act as a "JUMP UC,NAME3" order.

The entire process will now begin anew with the next execution of the SETSET order. All SETSET orders operate completely independently of each order.

The above four orders, if desired, may be used with PAX, provided only that the user defines them beforehand (see Appendix IV).

```
BOOFUN  MACRO   S1,A,S2,B,S3,C,FN
        VFD     3/C,15/S3,18/BOOFUN,3/B,15/S2,3/A,15/S1
        BCI     *,FN
        END
BOOLOP  MACRO   PLANE1,A,OP,PLANE2,B,PLANE3,C,BAR
        VFD     3/C,15/PLANE3,18/OP-BAR,3/B,15/PLANE2,3/A,15/PLANE1
        END                             OP=CLEAR,EQUAL,AND,OR,EXOR
BUBBLE  MACRO   P,KP,PLANE,K,CYCLES,KC
        VFD     3/KP,15/P,18/BUBBLE,3/KC,15/CYCLES,3/K,15/PLANE
        END
CLOCK   MACRO
                CLOCK
        END
COMMEN  MACRO   A
                COMMEN
        BCI     ,A
        END
CONNEC  MACRO   DL,K,PLANE1,A,PLANE2,B,PLANE3,C
        VFD     3/A,15/PLANE1,18/CONNEC,3/B,15/PLANE2,3/K,15/DL
        VFD     18/CONNEC+21,18/JUMP,3/C,15/PLANE3,18/24
        END
COORD   MACRO   X,Y
        VFD     18/X,18/Y
        END
CPRINT  MACRO   P,NN,MM,PLANE1,K1,PLANE2,K2,PLANE3,K3,PLANE4,K4,PLANE5,K5
        VFD     18/NN,18/PRINT,18/MM,3/P,18/K2,15/PLANE2,3/K3,15/PLANE3
        VFD     3/K4,15/PLANE4,3/K5,15/PLANE5,21/K1,15/PLANE1
        END
DL      MACRO   N,A,B,C,D,E,F,G,H,I
        DEC     N
        VFD     4/A,4/B,4/C,4/D,4/E,4/F,4/G,4/H,4/I
        END
DLTOIR  MACRO   DL,T,K,NAME
        VFD     3/T,15/DL,18/DLTOIR,18/NAME,18/K,50/0
        END
DOW     MACRO   WORD,K
        VFD     3/K,15/WORD,18/DOW
        END
HALT    MACRO   DUMP
                1902-DUMP
        END
INDEX   MACRO   OP,K,N                  OP=SET,INCR,DECR,TRANS
                OP,,K
        DEC     N
        END
J7094C  MACRO   A,T
        VFD     36/J7094C,21/T,15/A
        END
J7094M  MACRO   N
        VFD     36/J7094M,36/N
        END
JUMP    MACRO   OP,WHERE,TEST,C
        VFD     18/WHERE,18/OP,18/C,18/TEST
        END
LETTER  MACRO   A
                LETTER
        BCI     6,A
```

```
        END
LIST    MACRO   NAME,N
NAME    BSS     N
        END
LISTXY  MACRO   S1,T,LIST,N,U
        VFD     3/T,15/S1,18/LISTXY,3/U,15/N+1,18/LIST
        END
MARK    MACRO   DL,KD,PLANE1,A,PLANE2,B,PLANE3,C
        VFD     3/A,15/PLANE1,18/MARK,3/B,15/PLANE2,3/KD,15/DL
        VFD     18/MARK+25,18/JUMP,3/C,15/PLANE3,18/25
        END
PRINT   MACRO   PLANE,K,N,M
        VFD     18/N,18/PRINT,18/M,3/1,51/0,36/0,21/K,15/PLANE
        END
PRINTT  MACRO
                PRINTT
        END
PUNCH   MACRO   PLANE,P
        VFD     3/P,15/PLANE,18/PUNCH
        END
READ    MACRO   PLANE,P
        VFD     36/READ,21/P,15/PLANE
        END
READT   MACRO
                READT
        END
READZ   MACRO   ZWORD,XX,X,YY,Y,BIT,B,LIST,TAG
        VFD     3/B,15/BIT,18/READZ,3/X,15/XX,3/Y,15/YY,36/ZWORD
        IRP     LIST
        VFD     3/TAG,15/LIST
        IRP
        END
RENAME  MACRO   I,NAME
NAME    EQU     M*I
        END
SETMOD  MACRO   M
        VFD     36/SETMOD,36/M
        END
SHIFT   MACRO   PLANE,K,X,Y
        VFD     18/X,18/SHIFT,18/Y,3/K,15/PLANE
        END
SHIFTD  MACRO   PLANE,P,D,E,N,M
        VFD     3/E,15/D,18/SHIFTD,3/M,15/N,3/P,15/PLANE
        END
TCHAIN  MACRO   D,K,PLANE1,A,PLANE2,B,PLANE3,C,REL,N,X
        VFD     3/C,15/PLANE3,18/TCHAIN,3/B,15/PLANE2,3/A,15/PLANE1
        VFD     3/X,15/N,3/K,15/D,36/REL
        END
TIME    MACRO
                TIME
        END
TIMES   MACRO
                TIMES
        END
TMARK   MACRO   DLNAME,D,PLANE1,A,PLANE2,B,PLANE3,C,REL,N,K
        VFD     3/C,15/PLANE3,18/TMARK,3/B,15/PLANE2,3/A,15/PLANE1
        VFD     3/K,15/N,3/D,15/DLNAME,36/REL
```

```
        END
WRITE   MACRO   S1,T,X,XX,Y,YY
        VFD     3/T,15/S1,18/WRITE,3/XX,15/X,3/YY,15/Y
        END
WRITEP  MACRO   S1,T,NAME,S
        VFD     3/T,15/S1,18/WRITEP,21/S,15/NAME
        END
```

```
*       PAX     AN IBM 7094 (7090-COMPATIBLE) PROGRAM TO SIMULATE THE
*             OPERATION OF THE PATTERN ARTICULATION UNIT OF ILLIAC III
*
*
*
EAXM    OPD     076000120016
LAXM    OPD     476000120016
*
*
*
BOOLOP  MACRO   PLANE1,A,OP,PLANE2,B,PLANE3,C,BAR
        VFD     3/C,15/PLANE3,18/OP-BAR,3/B,15/PLANE2,3/A,15/PLANE1
        END                             OP=CLEAR,EQUAL,AND,OR,EXOR
BUBBLE  MACRO   P,KP,PLANE,K,CYCLES,KC
        VFD     3/KP,15/P,18/BUBBLE,3/KC,15/CYCLES,3/K,15/PLANE
        END
DL      MACRO   N,A,B,C,D,E,F,G,H,I
        DEC     N
        VFD     4/A,4/B,4/C,4/D,4/E,4/F,4/G,4/H,4/I
        END
INDEX   MACRO   OP,K,N                  OP=SET,INCR,DECR,TRANS
                OP,,K
        DEC     N
        END
J7094C  MACRO   A,T
        VFD     36/J7094C,21/T,15/A
        END
J7094M  MACRO   N
        VFD     36/J7094M,36/N
        END
JUMP    MACRO   OP,WHERE,TEST,C
        VFD     18/WHERE,18/OP,18/C,18/TEST
        END
MARK    MACRO   DL,KD,PLANE1,A,PLANE2,B,PLANE3,C
        VFD     3/A,15/PLANE1,18/MARK,3/B,15/PLANE2,3/KD,15/DL
        VFD     18/MARK+25,18/JUMP,3/C,15/PLANE3,18/25
        END
SETSET  MACRO   DL,K,ALPHA,BETA,GAMMA
        VFD     18/ALPHA,18/SETSET,18/BETA,3/K,15/DL
        VFD     18/GAMMA,18/JUMP-31,36/0
        END
SETTRA  MACRO   D,ALPHA
        VFD     36/SETTRA,18/ALPHA,18/D
        END
SETTRB  MACRO   K,ALPHA
        VFD     36/SETTRB,18/K,18/ALPHA
        END
SETTRC  MACRO   D,ALPHA
        VFD     36/SETTRC,18/ALPHA,18/D
        END
SHIFT   MACRO   PLANE,K,X,Y
        VFD     18/X,18/SHIFT,18/Y,3/K,15/PLANE
        END
        NOCRS
*
*
*
```

```
*       MASTER ROUTINE, EQUIVALENT TO CONTROL COUNTER
*
        EAXM
        AXC     ORDERS,7
MASTER  TRA*    ,7
RETTEN  TXI     *-1,7,-2
GOPAX   CLA     1,4
        PAC     ,7
        SXA     *+2,4
        TRA*    ,7
LPAX    LXA     **,4
        TRA     2,4
PAX     EQU     MASTER
*
*
*
*   MISC ROUTINES ' SET PLANE TRUE/COMPLEMENTED/NULL
*
SETCOM  AXT     ,2
        TRA     SETTRU+1
SETTRU  AXT     -1,2
        AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LDI     =014
        TSX     MODIFY,4
        CLA     MODSEL                  PUT MODE SIZE IN ACCUMULATOR
        PAX     ,1
        STA     D                       SAVE MODE SIZE TEMPORARILY
        ADM     ORDERM+1
        STA     D+2                     SET ADDRESS OF OBJECT PLANE
        ARS     18
        ADM     D
        STA     D                       SET ADDRESS OF ARGUMENT PLANE
D       CAL     **,1                    OBTAIN WORD OF ARGUMENT PLANE
        XEC     COMP,2                  COMPLEMENT IF NECESSARY
        SLW     **,1                    STORE WORD IN OBJECT PLANE
        TIX     *-3,1,1
        TXI     MASTER,7,-2             FINAL EXIT FROM SETCOM/SETTRU
COMP    CCM
        NOP
SETFUL  AXT     1,2
        TRA     *+2
SETNUL  AXT     ,2
        AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LDI     =014
        TSX     MODIFY,4
        CLA     MODSEL
        PAX     ,1
        ADM     ORDERM+1
        STA     *+3
        CLM
        XEC     COMP+1,2
        SLW     **,1
        TIX     *-1,1,1
```

```
        TXI     MASTER,7,-2          EXIT FROM SETNUL
*
*
*
*       MISC ROUTINES ' AND/OR PLANE TRUE/COMPLEMENTED
*
XORCOM  AXT     ,2
        TRA     *+2
XORTRU  AXT     -1,2
        AXT     -2,3
        TRA     ORTRUE+2
ANDCOM  AXT     ,2
        TRA     *+2
ANDTRU  AXT     -1,2
        AXT     ,3
        TRA     ORTRUE+2
CRCOMP  AXT     ,2
        TRA     *+2
CRTRUE  AXT     -1,2
        AXT     -1,3
        AXT     2,1
        TSX     MCRDER,4
        CLA     =144
        LDI     =015
        TSX     MCDIFY,4
        CLA     MODSEL               PUT MODE SIZE IN ACCUMULATOR
        PAX     ,1
        STA     E                    SAVE MODE SIZE TERMPRARILY
        ADM     ORDERM+1
        STA     E+3                  SET ADDRESS OF OBJECT PLANE
        ARS     18
        ADM     E
        STA     ANDCR,3              SET ADDRESS OF ARGUMENT PLANE
        CAL     ORDERM
        ARS     18
        ADM     E
        STA     E                    SET ADDRESS OF CONTEXT PLANE
E       CAL     **,1                 OBTAIN WORD OF CONTEXT PLANE
        XEC     COMP,2               COMPLEMENT IF NECESSARY
        XEC     ANDOR,3              COMBINE WITH ARGUMENT PLANE
        SLW     **,1                 STORE WORD IN OBJECT PLANE
        TIX     E,1,1
        TXI     MASTER,7,-2          FINAL EXIT
ANDOR   ANA     **,1                 AND ARGUMENT PLANE TO CONTEXT PLANE
        ORA     **,1                 OR ARGUMENT PLANE TO CONTEXT PLANE
        ERA     **,1                 EXCLUSIV OR ARG PLANE TO CONTXT PLANE
*
*
*
*       SUBROUTINE CONCERNING THE OPERATING MODE OF THE SIMULATOR
*
SETMOD  CLA     1,7
        STO     FXMODE
        PAX     ,4
        TXL     *+2,4,40
        ADD     1,7
        STO     MODSEL
```

```
        TXI     MASTER,7,-2
MODSEL  DEC     36
FXMODE  DEC     36
*
*
*
*       SHIFTD - SHIFT PLANE S1 IN THE D-DIRECTION N PLACES
*
SHIFTD  AXT     2,1
        TSX     MORDER,4
        CLA     =1
        LCI     =05
        TSX     MODIFY,4
        LXD     ORDERM+1,4          XR4=N (2'S COMPLEMENT IF N'S NEG.)
        PXA     ,4
        TXL     *+4,4,16384         JUMP IF N POSITIVE
        PAC     ,4
        PXA     ,4
        SSM
        STO     IR-1+21             ACCUMULATOR HAS N AS SIGNED INTEGER
        SXA     IR-1+22,7           SAVE XR7 IN IR22
        LXD     ORDERM,4            XR4=D
        SXA     *+3,4
        AXC     *+1,7
        TRA     SETTRA
        VFD     18/CRDERM,18/**     **=D BY *-3
        SETTRB  21,CRDERM
        J7094C  *+1
        LXA     IR-1+22,7
        TRA     SHIFT+2
*
*
*
*       SHIFT - ROUTINE TO PERFORM ARBITRARY TRANSLATION OF A PLANE
*
SHIFT   AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LCI     =01C
        TSX     MODIFY,4
        CLA     ORDERM
        ARS     18
        TZE     VSHFT               JUMP IF X=0
        TPL     *+5                 JUMP FOR RIGHT SHIFT
        PAC     ,1                  SET XR1 = ABS VAL OF -X
        PXA     ,1                  ACC = ABS VAL OF -X
        ACL     LGL
        TRA     *+2
        ACL     LGR
        SLW     SHIFTY              PLANT RIGHT/LEFT SHIFT INSTRUCTION
        CLA     FXMODE              FIND M
        PAX     ,1                  SET XR1 = M
        ADM     ORDERM+1            ACC ADDRESS IS PLANE+M
        STA     SHIFTY-1
        STA     SHIFTY+1
        LXA     FXMODE,4
        TXH     TIX+2,4,50
```

```
        CLA     TIX                     36X36 MODE
        STO     SHIFTY+2
        LGL     36                      CLEAR MQ
        TRA     SHIFTY-1
LGL     LGL     **                      **=ABSOLUTE VALUE OF X
LGR     LGR     **                      **=ABSOLUTE VALUE OF X
TIX     TIX     SHIFTY-1,1,1
        STQ     **,1                    **=PLANE+X+72
        ADD     =72                     AC ADDRESS IS  PLANE+X+72
        STA     TIX+1
        STA     SHIFTY-2
        CLA     TIX+1
        STO     SHIFTY+2
        LDQ     **,1                    **=PLANE+X+72
        CAL     **,1                    **=PLANE+X
SHIFTY          **                      SHIFT X PLACES RIGHT/LEFT
        SLW     **,1                    **=PLANE+X
                **                      TIX SHIFT-1,1,1  OR  STQ **,1
        TIX     SHIFTY-2,1,1
VSHFT   SLF
        CLA     FXMODE
        STO     COMMON                  STORE M=MODE DIMENSION (36/40/72)
        CLA     ORDERM+1
        STZ     COMMON+1
        STA     COMMON+1                STORE PLANE ADDRESS
FF      ARS     18
        TZE     RETTEN                  EXIT IF Y=0
        TMI     DSHFT                   JUMP FOR DOWN SHIFT (Y IS NEGATIVE)
USHFT   SLW     COMMON+2                STORE +Y
        ACL     COMMON+1                PLANE+Y IN ACCUMULATOR
        STA     MOVE
        CLA     COMMON+1                PLANE
        STA     MOVE+1
        ADD     COMMON                  PLANE+M
        STA     MOVE+4
        AXT     -1,4
        TRA     F
DSHFT   PAC     ,4                      COMPLEMENT Y
        PXA     ,4
        SLW     COMMON+2                STORE -(-Y)=ABS VAL OF Y
        ACL     COMMON+1                PLANE+Y IN ACC
        STA     MOVE+4
        CLA     COMMON+1
        ADD     COMMON
        SUB     =1                      PLANE+M-1
        STA     MOVE+1
        SUB     COMMON+2                PLANE+M-1-Y
        STA     MOVE
        AXT     1,4
F       SXD     MOVE+2,4                SET INCREMENTER POS/NEG
        AXT     ,2                      CLEAR XR2
        LXA     COMMON+2,4              SET XR4=Y
        CLA     COMMON
        SUB     COMMON+2                M-Y
        PAX     ,3                      SET XR3=M-Y
*                                       INITIAL SETTINGS FOR NEXT FIVE ORDERS
*                                           USHIFT    DSHIFT    BOTH
```

```
MOVE    CAL     **,2                    P+Y         P+M-1-Y    XR2=0
        SLW     **,2                    P           P+M-1
        TXI     *+1,2,**                -1          +1
        TIX     *-3,3,1                                        XR3=M-Y
        STZ     **,4                    P+M         P+Y        XR4=Y
        TIX     *-1,4,1
        CLA     FXMODE
        SUB     =36
        TNZ     *+2                     JUMP IF IN 72-MODE
        TXI     MASTER,7,-2
        CLA     COMMON+1
        ADD     =72
        STO     COMMON+1                SET NEW PSEUDO PLANE = PLANE+72
        SLT     1
        TRA     *+2
        TXI     MASTER,7,-2
        CLA     ORDERM+1
        SLN     1
        TRA     FF
*
*
*
*       PRINT - COMBINES UP TO 5 PLANES IN A SINGLE PICTORIAL PRINTOUT
*
PRINT   AXT     5,1
        TSX     MORDER,4
        CLA     =144
        LDI     =C1360
        TSX     MODIFY,4
        CLA     FXMODE                  FIND M
        STO     COMMON
        CLA     ORDERM
        PDX     ,1                      SET XR1=Y=LINE COUNTER
        ARS     18
        TSX     NMOK,4
        ADD     COMMON
        ARS     1                       ACC=(M+Y)/2
        STO     COMMON+1
        ADD     ORDERM+4
        STA     LDI+12                  STORE S1+(M+Y)/2
        CAL     ORDERM+2
        ADD     COMMON+1
        STA     LDI+6
        ARS     18
        ADD     COMMON+1
        STA     LDI+9
        CAL     ORDERM+3
        ADD     COMMON+1
        STA     LDI
        ARS     18
        ADD     COMMON+1
        STA     LDI+3
        CLA     ORDERM+1
        LGR     18
        TSX     NMOK,4
        STA     NXTBT-1                 STORE X, TO SET XR2
        ADD     COMMON
```

```
        ARS     1               ACC=(M+Y)/2
        ADD     XR7+1           ACC=B1ORO+(M+Y)/2
        STA     XEC
        SUB     =36
        STA     XEC+1
        PXA     ,0
        STD     WALL
        LGL     3               ACC=P
        STO     COMMON+2
        ALS     1
        ADD     COMMON+2
        PAX     ,3              SET XR3=3P
        TXI     *+1,7,-2        SKIP THE PSEUDO SCW AT ORDERS+2,7
        SLF
        CLA     FXMODE
        SUB     =36
        TZE     NXTLN-4         JUMP TO IGNORE 72-MODE CONSIDERATIONS
        CLA     ORDERM+1
        ARS     1               ACC DECREMENT HAS X/2
        STD     WALL+1
        PDX     ,4
        TXI     *+1,4,1
        SXD     WALL,4          WALL HAS X/2+1 IN DECREMENT
        SXA     XR7,7           SAVE XR7
        AXT     ,7
        CAL     =1
        TRA     *+2
NXTLN   CAL     =H
        SLW     COMMON+1        SET CARRIAGE CONTROL CHARACTER =1,BLK
        AXT     5,6             INITIAL SETTING  XR6=SEXTAD COUNTER
        AXT     -1,5            CLEAR XR5=WORD COUNTER
        AXT     **,2            SET XR2 = BIT/LINE COUNTER, **=X
NXTBT   PXA     ,0              CLEAR ACC
        TRA     *+16,3          XR3=3P, SKIP PLANES TO BE IGNORED
LDI     LDI     **,1            **=S5+(M+Y)/2
        XEC     XEC,7
        ADD     =16
        LDI     **,1            **=S4+(M+Y)/2
        XEC     XEC,7
        ADD     =8
        LDI     **,1            **=S3+(M+Y)/2
        XEC     XEC,7
        ADD     =4
        LDI     **,1            **=S2+(M+Y)/2
        XEC     XEC,7
        ADD     =2
        LDI     **,1            **=S1+(M+Y)/2
        XEC     XEC,7
        ADD     =1
        PAC     ,4
        CAL     COMMON,5
        ALS     6
        ADD     SYMBL,4
        SLW     COMMON,5
        TIX     *+3,6,1         JUMP UNLESS COMMON,5 IS FULL
        TXI     *+1,5,-1        INCREMENT WORD COUNTER
        AXT     6,6             RESET SEXTAD COUNTER
```

```
WALL    TXH     *+5,2,**            **=0 IN 36 MODE, **=X/2+1 IN 72 MODE,
        TXL     *+4,2,**            **=X/2, JUMP UNLESS LINE IS HALF DONE
        TXI     *+1,1,-72
        SLN     2
        TXI     *+1,7,-1
        TIX     NXTBT,2,1           JUMP FOR NEXT BIT IN ROW IF IT EXISTS
        SLT     2
        TRA     *+3                 JUMP TO IGNORE 72 MODE CONSIDERATIONS
        TXI     *+1,1,72
        AXT     ,7
        LCQ     =H
        CAL     COMMON,5
        LGL     6                   COMPLETE COMMON,5 WITH SPACES
        TIX     *-1,6,1
        SLW     COMMON,5
        TXI     *+1,5,-1
        PXD     ,5
        COM
        STD     *+2
        CALL    SYSWOT              (SYSWOT) - TEMPORARY
        TIX     COMMON+1,,**        **=L=NO. OF WORDS AT COMMON
        TNX     *+2,1,1             EXIT IF DONE
        TRA     NXTLN
        LXA     XR7,7
        TXI     MASTER,7,-3
NMOK    LBT
        CAS     FXMODE
        TRA     *+3
        TRA     1,4
        TRA     1,4
        AXT     3,1
        TRA     INDERR
XEC     XEC     **,2                **=BIORO+(M+X)/2
        XEC     **,2                **=BIORO+(M+X)/2-36
XR7                                 TEMP. STORAGE FOR XR7 AND FOR X/2+1
                BIORO
*
*
*
BIORO   LFT     400000
        LFT     200000
        LFT     100000
        LFT     040000
        LFT     020000
        LFT     010000
        LFT     004000
        LFT     002000
        LFT     001000
        LFT     000400
        LFT     000200
        LFT     000100
        LFT     000040
        LFT     000020
        LFT     000010
        LFT     000004
        LFT     000002
        LFT     000001
```

```
        RFT     4C0C00
        RFT     200000
        RFT     100C00
        RFT     040C00
        RFT     020C00
        RFT     010000
        RFT     004C00
        RFT     002C00
        RFT     001C00
        RFT     000400
        RFT     000200
        RFT     000100
        RFT     000C40
        RFT     000C20
        RFT     000C10
        RFT     000C04
        RFT     000C02
        RFT     000001
*
*
*
*       LETTER - THIS ROUTINE SETS THE PRINT CHARACTERS OF 'PRINT'
*
LETTER  AXT     5,1
        AXT     ,2
        AXT     6,3
        LDQ     1,7
        CLM
        LGL     6
        SLW     SYMBL,2
        TXI     *+1,2,-1
        TIX     *-4,3,1
        TXI     *+1,7,-1
        TIX     *-8,1,1
        CLM
        LDQ     1,7
        LGL     6
        SLW     SYMBL,2
        CLM
        LGL     6
        SLW     SYMBL+1,2
        TXI     MASTER,7,-2
SYMBL   OCT     60,1,2,21,3,22,23,24,4,25,26,27,30,31,41,42,5,43,44,45
        OCT     46,47,50,51,62,63,64,65,66,67,70,71
*
*
*
*       -READT- AND -PRINTT- = ROUTINES TO READ, PRINT PATTERN TITLES
*
READT   CALL    SYSRIT
        TIX     PATLBL,,LABELT
        CLA     PATLBL
        CAS     =HTITLE
        TRA     *+2
        TXI     MASTER,7,-1
LABELT  AXT     5,1
        TRA     INDERR
```

```
PRINTT CALL    SYSHOT
       TIX     PATLBL+1,,11
       TXI     MASTER,7,-1
*
*
*
*      BUBBLE - THIS ROUTINE BUBBLES P CONSECUTIVE PLANES FOR C CYCLES
*
BUBBLE AXT     2,1
       TSX     MORDER,4
       CLA     =144
       LDI     =010
       TSX     MODIFY,4
       CLA     =1
       LDI     =05
       TSX     MODIFY,4
       CLA     MODSEL
       PAX     ,4                      SET XR4=M'
       CAL     ORDERM+1
       STD     COMMON+1                SAVE C
BUBB   STA     BUB
       STA     BUB+1
       STA     BUB+3
       STA     BUB+9
       ADD     =144
       STA     BUB+8
       LXD     COMMON+1,3              SET XR3=C
       CLA     ORDERM
       PDX     ,2                      SET XR2=P
       CAL     =0777777777777
       SLW     COMMON                  SET TOP BOUNDARY CONDITION
       AXT     ,1                      CLEAR XR1 TO START CYCLE ON TOP PLANE
BUB    LDQ     **,1                    **=PLANE
       CAL     **,1                    **=PLANE
       ANA     COMMON                  FORM A TIMES B
       SLW     **,1                    **=PLANE
       TNX     RSBUB,2,1               JUMP IF ROW OF LAST PLANE IS STORED
       XCL
       SLW     COMMON                  STORE B, THIS WILL BE A ON NEXT CYCLE
       CCM
       ANA     **,1                    **=PLANE+144, FORM B-BAR TIMES C
       ORS     **,1                    **=PLANE, STORE (AB)OR(B-BAR.C)
       TXI     BUB,1,-144              FORM NEW ADDRESSES, B BECOMES A, ETC
RSBUB  TIX     BUB-5,3,1               JUMP TO START NEW CYCLE ON SAME ROW
       CLA     BUB                     NO JUMP, START ON NEW ROW
       ADD     =1
       TIX     BUBB,4,1                JUMP IF M' ROWS ARE NOT YET COMPLETED
       TXI     MASTER,7,-2             FINAL EXIT
*
*
*
*      COMMEN - A SUBROUTINE TO PRINT SHORT CODE CHECK COMMENTS
*
COMMEN AXT     11,1
       TSX     MORDER,4
       CALL    SYSHOT
       TIX     ORDERM+1,,10
```

```
        TXI     MASTER,7,-11
*
*
*
*       INDERR - THIS INTERNAL ROUTINE FLAGS PROGRAMMING ERROR, THEN
*         TRANSFERS TO SYSERR, TERMINATING THE SIMULATOR PROGRAM.
*
INDERR  PXA     ,1
        ADM     =H-ERR 0
        STO     COMMON
        CALL    SYSWOT
        TIX     COMMON,,1
        CALL    SYSERR
*
*
*
*       THE NEXT 2 ROUTINES TRANSFER BETWEEN 7090 AND PAU-SIMULATOR
*
J7094C  TXI     *+1,7,-2
        TRA*    -1,7
J7C94M  TXI     *+1,7,-2
        PXA     ,7
        PAX     ,1
        SUB     -1,7
        PAX     ,7
        TRA     ,1
*
*
*
*       READ - READS PUNCHED INPUT PATTERNS (PICTORIAL OR BINARY)
*
READ    AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LDI     =01C
        TSX     MODIFY,4
        LAXM
        NOP
        ZET     7                       L(7)=0 MEANS  INPUT  IS  BCD
        TRA     RBCDS
        CALL    .READ
        STR     FORM1                   SPEC IS 2I2,A2,11A6
        STR     COMMON,,COMMON+2
        STR     PATLBL+1,,PATLBL+11
        STR
        EAXM
        CLA     FXMODE
        STO     COMMON+3
        SUB     COMMON+1                ACC=M-X
        TMI     2BIG                    JUMP IF M LESS THAN X
        ARS     1                       ACC=(M-X)/2
        SUB     =36
        STA     OLOOP-1                 ACC ADDRESS IS 36-(M-X)/2
        CLA     COMMON+3
        SUB     COMMON                  ACC=M-Y
        TPL     *+6                     JUMP UNLESS M LESS THAN Y
        TRA     *+3
```

```
        LBT
        TRA     1,4
2BIG    AXT     1,1
        TRA     INDERR
        ARS     1                       ACC=(M-Y)/2
        PAX     ,1                      XR1=(M-Y)/2, CLEARING COUNTER
        LXA     COMMON,5                XR5=Y, ROW COUNTER
        ADD     ORDERM+1
        STA     ZZZST                   STORE PLANE+(M-Y)/2
        ADD     =72
        STA     72CLR+1                 STORE PLANE+72+(M-Y)/2
        CLA     COMMON+1
        TSX     2BIG-2,4
        CLA     COMMON
        TSX     2BIG-2,4
        ADD     COMMON+3
        ARS     1                       ACC=(M+Y)/2
        ADD     ORDERM+1                ACC=PLANE+(M+Y)/2
        AXT     ,2
        LDI     FXMODE
        RFT     44
        TRA     *+4                     JUMP IF IN 36 MODE
        AXT     -1,2
        STA     ACCFL
        ADD     =72
        STA     STORE+3                 STORE PLANE+(M+Y)/2+(EITHER 0 OR 72)
        CLA     ORDERM+1
        ADD     COMMON+3                ACC HAS PLANE+M
        STA     ZZZST+1
        ADD     =72                     ACC HAS PLANE+M+72
        STA     ZZZST+3
        CLA     COMMON+2
        SUB     AID
        STO     COMMON+2
        TXL     *+6,1,0
ZZZST   STZ     **,1                    **=PLANE+(M-Y)/2
        STZ     **,1                    **=PLANE+M
        XEC     72CLR,2
        STZ     **,1                    **=PLANE+72+M
        TIX     *-4,1,1
        AXT     1,6                     CLEAR XR6 FOR USE IN ID CHECK
INPUU   LAXM
        NOP
        CALL    .READ
        STR     FORM2
        STR     COMMON+3,,COMMON+14
        NZT     COMMON+2                SKIP FOR NO ID CHECK
        TRA     CHKID
IDRET   STR                             END OF I/O LIST
        EAXM
        AXT     ,1                      CLEAR WORD COUNTER
        CLM
        LXA     COMMON+1,3              SET XR3=X
        AXT     **,4                    SET XR4=**=36-(M-X)/2
CLOOP   LDI     COMMON+3,1              GET WORD OF BCD CHARACTERS
        IIS     =H
        AXT     6,2                     SET XR2=6=SEXTAD COUNTER
```

```
ILOOP   XEC     TSTSTI+6,2
        ADD     =1
        TNX     STORE,3,1               JUMP IF X BITS ARE READ
        TNX     ACCFL,4,1               JUMP IF ACC IS FULL
        ALS     1
6BITS   TIX     ILOOP,2,1               JUMP IF 6 BITS (ONE WORD) NOT YET REA
        TXI     OLOOP,1,-1
ACCFL   SLW     **,5                    **=PLANE+(M+Y)/2
        AXT     36,4
        TRA     6BITS-1
STORE   TNX     *+3,4,1                 JUMP IF ACC IS FULL
        ALS     1
        TRA     *-2
        SLW     **,5                    IN 36 MODE, **=PLANE+(M+Y)/2
*                                       IN 72 MODE, **=PLANE+(M+Y)/2+72
        TIX     INPUU,5,1               JUMP FOR NEXT ROW
        TXI     MASTER,7,-2
72CLR   TRA     ZZZST+4
        STZ     **,1                    **=PLANE+72+(M-Y)/2
AID     BCI     1,ID
FORM1   BCI     2,2I2,A2,11A6*
FORM2   BCI     2,12A6,S6,I2*
CHKID   STR     COMMON
        EAXM
        PXA     ,6                      ACC=CURRENT VALUE OF ID NUMBER
        SUB     COMMON
        TZE     *+3                     JUMP IF CARD SEQUENCE IS OK
        AXT     2,1                     CARD SEQUENCE IS INCORRECT,
        TRA     INDERR                    INDICATE PROGRAMMING ERROR 2
        TXI     *+1,6,1
        LAXM
        NOP
        TRA     IDRET
TSTSTI  LFT     770000
        LFT     007700
        LFT     000077
        RFT     770000
        RFT     007700
        RFT     000077
RBCDS   LAC     ORDERM+1,2              XR2 HAS LOADING ADDRESS COMPLEMENTED
        CALL    SYSRIT                  READ A BINARY CARD
        TIX     COMMON
        LXA     COMMON,1                XR1 HAS NO. OF WORDS ON CARD
        AXT     -1,4
        CLA     COMMON,4
        STO     ,2
        TXI     *+1,4,-1
        TXI     *+1,2,-1
        TIX     *-4,1,1
        LDI     COMMON
        RNT     700000                  IF T=7, THIS WAS LAST CARD, SKIP NEXT
        TRA     RBCDS+1
        EAXM
        TXI     MASTER,7,-2
*
*
*
```

```
*       PUNCH   THIS ROUTINE PUNCHES A BINARY PATTERN ONTO BINARY CARDS,
*       THE SIZE (36X36 OR 72X72) BEING DETERMINED BY THE OPERATING MODE.
*
PUNCH   AXT     1,1
        TSX     MORDER,4
        CLA     =144
        LDI     =1
        TSX     MODIFY,4
        CLA     MODSEL
        PAX     ,6                      XR6=N
PUNCJ   LXD     ORDERM,2                XR2=PLANE=S1=FWA
        AXT     23,5                    XR5=23=NO. OF WORDS/CARD (USUALLY)
        TIX     PUNCI,6,23              JUMP IF NEXT PUNCHED CARD IS NOT LAST
        PXA     ,6                      PREPARE TO PUNCH LAST CARD, WHICH MAY
        PAX     ,5                         HAVE FEWER THAN 23 WORDS ON IT.
        TSX     ECARD,1
        TXI     MASTER,7,-1             PUNCHING COMPLETE, EXIT
PUNCI   TSX     PCARD,1                 GO PUNCH 1 CARD, NOT THE LAST ONE
        TXI     PUNCJ+1,2,23            ADVANCE FWA, GO PUNCH NEXT CARD
*
*               ENTER 'PCARD' VIA 'TSX PCARD,1', WITH XR5=N, XR2=FWA.
*       THEN WORDS  FWA  THROUGH  FWA+N-1  WILL BE PUNCHED IN BINARY CARD
*       COLUMNS 4 THROUGH 72(MAX.).  COLUMNS 1-3 CONTAIN P,D,T,A, WHERE
*       P=T=0, D  IS BINARY CARD IDENTIFICATION, AND A=N (N MAX. IS 23).
*               'ECARD' IS THE SAME AS 'PCARD', EXCEPT  T=7 (LAST CARD)
*
ECARD   CAL     =0500700000
        TRA     *+2
PCARD   CAL     =0500000000
        XCL                             PART OF OVERWRITING WORD IS NOW IN MQ
        SXA     COMMON,2                SAVE FWA
        TXI     *+1,2,-1                XR2=FWA-1=OVERWRITING ADDRESS
        PXA     ,2
        PAC     ,2
        STA     PCARE                   SET FWA PARAMETER FOR SYSPCB
        TXI     *+1,5,1                 XR5=N+1
        SXD     PCARE,5
        LDI     ,2                      SAVE WORD TO BE OVERWRITTEN
        XCL
        STO     ,2                      OVERWRITE TAG, DECREMENT, AND PREFIX
        TXI     *+1,5,-1                RESTORE XR5=N+1-1=N
        PXA     ,5
        STA     ,2                      OVERWRITE ADDRESS (=N)
        LAXM
        NCP
        CALL    SYSPCB
PCARE   TIX     **,,**                  **=FWA-1,N+1
        EAXM
        STI     ,2                      RESTORE OVERWRITTEN WORD
        LXA     COMMON,2                RESTORE XR2=FWA
        TRA     1,1                     LINK RETURN
*
*
*
*       THE FOLLOWING SUBROUTINES PERFORM INDEX REGISTER ARITHMETIC
*
DECREM CLA      ,7
```

```
        PDC     ,1                      XR1 GETS COMPLEMENT OF IR NUMBER
        CLA     1,7
        CHS
        TRA     INCREM+3
SETIR   CLA     ,7
        PDC     ,1
        STZ     IR-1,1                  CLEAR IRK FOR USE BY INCREM
INCREM  CLA     ,7
        PDC     ,1
        CLA     1,7                     OBTAIN SIGNED C IN ACC
        ADD     IR-1,1
        TNZ     *+2                     ARE NEW CONTENTS OF IR ZERO$
        SSP                               YES, DISALLOW MINUS ZERO
        STO     IR-1,1                    NO
        PBT                             TEST FOR IR OVERFLOW
        TXI     MASTER,7,-2
        AXT     4,1                       OVERFLOW, INDICATE ERROR NUMBER 2
        TRA     INDERR
TRANS   CLA     ,7
        PDC     ,1
        CLA     1,7
        PAC     ,2
        CLA     IR-1,2
        STO     IR-1,1
        TXI     MASTER,7,-2
DSPLY   CLA     ,7
        PDC     ,1                      XR1 GETS COMPLEMENT OF IR NUMBER
        ARS     18
        STO     ORDERM
        CLA     IR-1,1
        STO     ORDERM+1                STORE CONTENTS OF IRK
        LAXM
        NOP
        CALL    .PRINT
        STR     *+5
        STR     ORDERM,,ORDERM+1
        STR
        EAXM
        TXI     MASTER,7,-2
        BCI     4,3H-IR,I2,3H = ,I12*
*
*
*
*       THE FOLLOWING ROUTINE EFFECTS SYMBOL MODIFICATION BY INDEX REGS
*
MORDER  PXA     ,7
        PAX     ,6
        SXA     COMMON+4,2              SAVE XR2, XR3 IN COMMON+4
        SXD     COMMON+4,3
        SXD     COMMON,1                SAVE W IN DECREMENT OF COMMON
        AXT     ,3
        CLA     ,6
        STO     ORDERM,3
        TXI     *+1,6,-1
        TXI     *+1,3,-1
        TIX     *-4,1,1                 JUMP IF SCw NOT YET FULLY TRANSMITTED
        TRA     1,4
```

```
        STP     COMMON+2
        PDC     ,5
        ALS     18
        STD     ORDERM,3
        TXI     MODIFY+9,2,1
        STT     COMMON+2
        PAC     ,5
        NOP
        STA     ORDERM,3
        TXI     MODIFY+12,6,1
MODIFY  STO     COMMON+1            STORE R
        AXT     ,2
        AXT     ,3
        AXT     1,6
        CLA     COMMON              ACC DECREMENT HAS W
        STD     *+7
        XEC     BIORO+35,2          SKIP FOR NO MODIFICATION (DECREMEN)
        TRA     MODDEC
        TXI     *+1,2,1
        XEC     BIORO+35,2          SKIP FOR NO MODIFICATION (ADDRESS)
        TRA     MODADD
        TXI     *+1,6,1
        TXH     *+3,6,**
        TXI     *+1,2,1
        TXI     *-8,3,-1
        LXA     COMMON+4,2          RESTORE XR2, XR3
        LXD     COMMON+4,3
        TRA     1,4
MODDEC  AXT     10,1
        TRA     *+2
MODADD  AXT     5,1
        STZ     COMMON+2
        CAL     ORDERM,3            OBTAIN WORD TO MODIFY
        XEC     MODIFY,1            COMMON+2 HAS K IN TAC OR PREFIX
        CAL     COMMON+2
        ARS     15                  ACC HAS K IN ADDRESS OR DECREMENT
        XEC     MODIFY+1,1          STORE -K IN XR5
        TXL     *+7,5,0
        LDQ     IR-1,5              V(IRK) TO MQ
        MPY     COMMON+1            V(IRK)R IN MQ
        XCA
        XEC     MODIFY+2,1          ACC HAS V(IRK) IN ADD OR DEC
        ADM     ORDERM,3
        XEC     MODIFY+3,1          STORE MODIFICATION
        XEC     MODIFY+4,1          RETURN
*
*
*
*       JUMPS ' SUBROUTINES TO TRANSFER CONTROL WITHIN SIMULATOR
*
JUMPUC  CLA     ,7
        PDC     ,7
        TRA*    ,7
JUMPNL  CLA     1,7
        TSX     NULIST,3
        TZE     JUMPUC
        TXI     MASTER,7,-2
```

```
JUMPNN CLA     1,7
       TSX     NULTST,3
       TNZ     JUMPUC
       TXI     MASTER,7,-2
JUMPZE TSX     SELIR,4
       ZET     IR-1,1
       TXI     MASTER,7,-2
       TRA     JUMPUC
JUMPNZ TSX     SELIR,4
       ZET     IR-1,1
       TRA     JUMPUC
       TXI     MASTER,7,-2
JUMPET TSX     TESTC,3
       TXI     MASTER,7,-2
       TRA     JUMPUC
       TXI     MASTER,7,-2
JUMPLT TSX     TESTC,3
       TRA     JUMPUC
       TXI     MASTER,7,-2
       TXI     MASTER,7,-2
JUMPGT TSX     TESTC,3
       TXI     MASTER,7,-2
       TXI     MASTER,7,-2
       TRA     JUMPUC
JUMP   TSX     SELIR,4
       PXA     ,7
       STA     IR-1,1
       TRA     JUMPUC
JUMPRS TSX     SELIR,4
       CLA     IR-1,1
       PAX     ,7
       TXI     MASTER,7,-2
TESTC  TSX     SELIR,4
       TPL     *+4                    JUMP IF C IS POSITIVE
       PCC     ,4
       PXD     ,4
       SSM
       ARS     18                     ACCUMULATOR HAS C AS FXD POINT INTEG
       CAS     IR-1,1
       TRA     1,3                    IRK IS LESS THAN C
       TRA     2,3                    IRK EQUALS C
       TRA     3,3                    IRK IS GREATER THAN C
SELIR  CLA     1,7                    THREE WORD ROUTINE TO SELECT AN INDE
       PAC     ,1                       REGISTER.  XR1 IS SET = COMPLEMENT
       TRA     1,4                      OF K, FOUND IN ADDRESS OF THE SCW.
*
*
*
*      NULTST ' AN INTERNAL SUBROUTINE TO TEST FOR NULL PLANE
NULTST STA     G
       ACL     =72
       STA     COMMON
GG     CLA     FXMODE
       PAX     ,4                     SET XR4=M=MODE DIMENSION (36/40/72)
       ADD     G
       STA     G                      ACCUMULATORS ADDRESS IS PLANE+M
G      ZET     **,4                   TEST WORD OF PLANE, **=PLANE+M
```

```
        TRA     1,3                     EXIT WITH NON-ZERO ACCUMULATOR
        TIX     G,4,1
        CLA     FXMODE
        SUB     =36
        TZE     GGG
        CAL     COMMON
        TZE     1,3                     NON-36 MODE, EXIT WITH ZERO ACCUMULTR
        STZ     COMMON                  NON'36 MODE, PLANE PARTIALLY TESTED
        STA     G                       SET PSEUDO'PLANE ADDRESS=PLANE+72
        TRA     GG
GGG     PXA     ,0                      CLEAR ACCUMULATOR
        TRA     1,3                     36X36 MODE, EXIT WITH ZERO ACCUMULATR
*
*
*
*       THE FOLLOWING ROUTINES ARE USED TO TIME SIMULATOR SUBROUTINES
*
        MZE                             MINUS SIGN DENOTES FIRST ENTRY
TIME    CLA     *-1
        TMI     TIMSET
        LXA     TIME-1,6
        CLA     5
        SUB     INEZ
        STO     IwW,6
        ADD     INEZ
        STO     INEZ
        TXI     *+1,6,-1
        SXA     TIME-1,6
        TXI     MASTER,7,-1
TIMSET  CLA     5
        STO     INEZ
        STZ     TIME-1
        TXI     MASTER,7,-1
TIMES   CAL     TIME-1
        COM
        SUB     *+10
        ALS     18
        STD     *+8
        LAXM
        NOP
        CALL    .PRINT
        STR     EJECT,,0
        STR
        CALL    .PRINT
        STR     TT,,0
        STR     IwW,,**
        STR
        EAXM
        SSM
        STO     TIME-1
        TXI     MASTER,7,-1
EJECT   BCI     1,1H1*
TT      BCI     6,1H ,I10,22H SIXTIETHS OF A SECOND*
INEZ
CLOCK   LAXM
        NOP
        STZ     5
```

```
        CALL    .COMNT
        STR     *+5,,0
        STR
        EAXM
        HPR
        TXI     MASTER,7,-1
        BCI     7,36HOPERATOR, TURN CLOCK ON, PRESS START*
*
*
*
*       DOW  -  DISPLAY OCTAL WORD
*
DOW     AXT     1,1
        TSX     MORDER,4
        CLA     =1
        LDI     =1
        TSX     MODIFY,4
        LXD     ORDERM,4
        SXA     *+5,4
        LAXM
        NOP
        TSX     .PRINT,4
        STR     *+5
        STR     **
        STR
        EAXM
        TXI     MASTER,7,-1
        BCI     4,11H-OCTAL WORD,K14*
*
*
*
*       THE FOLLOWING DIRECTION LISTS ARE MACHINE-DEFINED
*
DL0     DL      1,0
DL1     DL      1,1
DL2     DL      1,2
DL3     DL      1,3
DL4     DL      1,4
DL5     DL      1,5
DL6     DL      1,6
DL7     DL      1,7
DL8     DL      1,8
DL15    DL      2,1,5
DL26    DL      2,2,6
DL37    DL      2,3,7
DL48    DL      2,4,8
DLSUR   DL      8,1,2,3,4,5,6,7,8
DLALL   DL      9,0,1,2,3,4,5,6,7,8
*
*
*
*         THE NEXT 4 ROUTINES CONTROL X,Y PLANTING IN PTRANS SCW'S
*
*       SETTRA - SET THE SHIFT ORDER AT LOC ALPHA TO SHIFT IN D DIRECTION
*
SETTRA  CLA     1,7
        PAC     ,1                      XR1=-D
```

```
        CLA     1,7
        PDC     ,2                      XR2=-ALPHA
        CAL     DIRECT,1                ACC HAS X,Y VALUES FOR PLANTING
        STP     ,2
        STD     ,2
        ALS     18
        STP     1,2
        STD     1,2
        TXI     MASTER,7,-2
DIRECT  VFD     18/+0,18/+0
        VFD     18/+1,18/+0
        VFD     18/+1,18/+1
        VFD     18/+0,18/+1
        VFD     18/-1,18/+1
        VFD     18/-1,18/+0
        VFD     18/-1,18/-1
        VFD     18/+0,18/-1
        VFD     18/+1,18/-1
*
*
*
*       SETTRB - MULTIPLY THE X,Y OF SHIFT ORDER AT LOC ALPHA BY V(IRK)
*
SETTRB  CLA     1,7
        PAC     ,1                      XR1=-ALPHA
        PDC     ,2                      XR2=-K
        LDQ     ,1
        TSX     MULT,4
        STP     ,1
        STD     ,1
        LDQ     1,1
        TSX     MULT,4
        STP     1,1
        STD     1,1
        TXI     MASTER,7,-2
MULT    LGR     18
        MPY     IR-1,2
        XCA
        ALS     18
        TPL     *+2
        COM
        TRA     1,4
*
*
*
*       SETTRC - SET THE SHIFT ORDER AT LOC ALPHA TO SHIFT IN -D DIRECT'N
*
SETTRC  CLA     1,7
        PAX     ,1                      XR1=D
        TXL     *+4,1,0                 JUMP IF D=0
        TXI     *+1,1,4                 XR1=D+4
        TXL     *+2,1,8                 JUMP IF D+4 IS LESS OR EQUAL TO 8
        TXI     *+1,1,-8
        PXA     ,1                      ACC ADDRESS IS -D
        TRA     SETTRA+1
*
*
```

```
*
*       SETSET - SET THE SETTRA ORDERS AT LOCATIONS ALPHA,BETA WITH THE
*                VALUES D,-D, FOR EVERY D IN DL.  WHEN DL IS GONE, JUMP
*                TO LOCATION GAMMA
*
SETSET  AXT     2,1
        TSX     MORDER,4
        CLA     =2
        LDI     =01C
        TSX     MODIFY,4
        CLA     ORDERM
        PDC     ,1                      XR1=-ALPHA
        CLA     ORDERM+1
        PDC     ,2                      XR2=-BETA
        PAC     ,3                      XR3=-DL
        CLA     3,7
        ADD     =1
        CAS     ,3
        TRA     SETSE                   BEGIN FINAL EXIT (KTH ENTRY)
        NOP
        STO     3,7
        ALS     2                       ACC=4I
        LDQ     1,3
        STA     *+1
        LGL     **
        ANA     =15
        STA     1,1
        TZE     *+5                     JUMP IF D=0
        ADD     =4                      ACC=D+4
        CAS     =8
        SUB     =8
        NOP                             ACC=(D+4)MODULO 8
        STA     1,2
        TXI     MASTER,7,-4             NORMAL EXIT
SETSE   STZ     3,7
        TXI     MASTER,7,-2
*
*
*
*       DLTOIR - READ A DIRECTION LIST, PUT NEXT DIRECTION INTO IRK.
*                IF DIRECTION LIST IS EXHAUSTED, JUMP TO LOCATION NAME.
*
DLTOIR  CLA     2,7
        TNZ     DLTCIS                  JUMP IF THIS IS NOT INITIAL EXECUTION
        AXT     1,1
        TSX     MORDER,4
        CLA     =2
        LDI     =1
        TSX     MODIFY,4
        LDC     ORDERM,4                XR4= -DL
        CLA     1,4
        STO     3,7
        CLA     ,4
        TZE     DLTCIS+2                FINAL EXIT IF INPUT DL IS VACUOUS
DLTOIT  STO     2,7
        CLA     1,7
        PAC     ,4                      XR4=K
```

```
        ZAC
        LDQ     3,7
        LGL     4
        STQ     3,7
        STO     IR-1,4              PUT D (FROM DL) INTO IRK
        TXI     MASTER,7,-4         NORMAL EXIT
DLTOIS  SUB     =1
        TNZ     DLTOIT              JUMP IF MORE D'S ARE STILL IN DL
        STO     2,7
        CLA     1,7
        TRA     JUMPUC+1            FINAL EXIT IS JUMP UC TO 'NAME'
*
*
*
*     LISTXY - THIS ROUTINE LISTS THE POINTS IN A PLANE S1.  ONLY THE
*     FIRST N POINTS ARE LISTED.  PAX XR21 WILL CONTAIN M, THE NUMBER OF
*     POINTS FOUND, IF M LESS OR EQUAL TO N.  OTHERWISE M=N+1
*
LISTXY  AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LDI     =1
        TSX     MODIFY,4
        CLA     =1
        LDI     =4
        TSX     MODIFY,4
        CLA     MODSEL
        PAX     ,6
        LXD     ORDERM,4            XR4 = PLANE
        TXH     LIS7,6,50           JUMP TO LIS7 IF IN 72-MODE
        AXT     ,6                  SET MODE SWITCH = XR6 = 0
        TXI     *+1,4,35            SET XR4 = PLANE + 35 = BASE ADDRESS
LISS    SXA     LISU,4              SET PLANE BASE ADDRESS
        SXA     LISU+2,4            SET PLANE BASE ADDRESS
        LXD     ORDERM+1,5          XR5=N+1=N'
        TXH     *+2,5,1
        AXT     -1,5
        LXA     ORDERM+1,4          XR4=LIST
        SXD     *+1,5
        TXI     *+1,4,**            **=N' BY *-1, XR4 = LIST+N'
        SXA     CALCXY+2,4          STORE LIST+N'
        SXA     CALCXZ,4
        SXA     CALCXZ+4,4
        AXT     ,1                  CLEAR SR1 = Y-COORDINATE COUNTER
        STZ     IR-1+21             CLEAR PAX INDEX REGISTER 21
LISU    NZT     **,1                **= BASE ADDRESS OF PLANE
        TRA     LISQ+2              JUMP, WORD OF PLANE IS EMPTY
        LDI     **,1                **= BASE ADDRESS OF PLANE
        LFT     777777
        TRA     LISL
LISQ    RFT     777777
        TRA     LISR
        XEC     *+2,6
        TRA     *+5                 JUMP FROM HERE IF IN 72-MODE (XR6=1)
        TRA     *+1                 JUMP FROM HERE IF IN 36-MODE (XR6=0)
        TXI     *+1,1,1
        TXL     LISU,1,35
```

```
        TXI     MASTER,7,-2         FINAL EXIT (NORMAL, 36-MODE)
        TXL     *+4,1,71            JUMP IF JUST DONE WITH LEFT-HALF WORD
        TXI     *+1,1,73
        TXL     LISU,1,71
        TXI     MASTER,7,-2         FINAL EXIT ( NORMAL, 72-MODE )
        TXI     LISU,1,-72
LISL    LFT     777000
        TRA     LISLL
        LFT     777
        TRA     LISLM
        TRA     LISQ
LISR    RFT     777000
        TRA     LISRM
        RFT     777
        TRA     LISRR
        TRA     LISQ+2
LIS7    AXT     1,6                 SET XR6 = 1 = 36/72 MODE SWITCH
        TXI     LISS,4,71           SET XR4 = BASE ADDRESS OF PLANE
LISLL   AXT     ,2
        TSX     LOOKY,4
        NOP
        TRA     LISL+2
LISLM   AXT     9,2
        TSX     LOOKY,4
        ALS     9
        TRA     LISQ
LISRM   AXT     18,2
        TSX     LOOKY,4
        ALS     18
        TRA     LISR+2
LISRR   AXT     27,2
        TSX     LOOKY,4
        ALS     27
        TRA     LISQ+2
LOOKY   PIA
        XEC     1,4
        AXT     9,3
        PBT
        TXI     *+2,2,1             INCREMENT X COORDINATE
        TRA     CALCXY              JUMP FROM HERE IF P=1, CALCULATE X,Y
        ALS     1
        TIX     *-4,3,1
        TRA     2,4                 RETURN
CALCXY  TNX     XYOVF,5,1           JUMP IF THIS IS N+1TH POINT
        TXI     *+1,5,1
        STZ     **,5                **=LIST+N'
        STO     COMMON              SAVE ACCUMULATOR
        PXA     ,1
        TXL     CALCXZ,1,75
        TXI     *+1,1,72
        PXA     ,1
        TXI     *+1,1,-72
CALCXZ  STA     **,5                STORE Y COORDINATE, **=LIST+N'
        PXD     ,2
        TXL     *+2,1,71            SKIP NEXT INSTRUCTION IF IN LEFT HALF
        ADD     36SPEC
        STD     **,5                STORE X COORDINATE, **=LIST+N'
```

```
        CLA     IR-1+21
        ADD     =1
        STO     IR-1+21
        CLA     COMMON                  RESTORE ACCUMULATOR
        TXI     LOOKY+4,5,-1
36SPEC          ,,36
XYOVF   CLA     IR-1+21
        ADD     =1
        STO     IR-1+21
        TXI     MASTER,7,-2             FINAL EXIT  ( ON LIST OVERFLOW )
*
*
*
*       READZ   THIS ROUTINE READS P(I) FOR SEVERAL PLANES M(I)
*
READZ   AXT     21,1
        TSX     MORDER,4
        AXT     3,4
        SXD     COMMON,4                MODIFY THE MODIFY ROUTINE
        CLA     =1
        LDI     =015
        TSX     MODIFY,4
        LXA     FXMODE,4
        TXI     *+1,4,-1
        SXD     *+4,4
        SXD     *+5,4
        AXT     6,1
        LXD     ORDERM+1,4
        TXH     INDERR,4,**
        LXA     ORDERM+1,4
        TXH     INDERR,4,**
        AXT     -3,4
        SXA     MODIFY+2,4              MODIFY THE MODIFY ROUTINE
        AXT     18,4
        SXD     COMMON,4                MODIFY THE MODIFY ROUTINE
        SXD     MODIFY-6,0              MODIFY THE MODIFY ROUTINE
        CLA     =144
        LDI     =0777777
        TSX     MODIFY,4
        SXA     MODIFY+2,0              RESTORE THE MODIFY ROUTINE
        AXT     1,4
        SXD     MODIFY-6,4              RESTORE THE MODIFY ROUTINE
        LDC     ORDERM,1                XR1 HAS BIT NUMBER (COMPLEMENTED)
        CAL     =0400000000000
        ARS     ,1                      ACC HAS 'BIT' MASK (BMASK)
        XCL                             PUT BMASK INTO MQ
        CLA     ORDERM+2
        STA     READB
        STA     READB+2
        STA     READB+4
        TSX     XY,3
        STO     COMMON
        AXT     -3,6
READA   LDI     ORDERM,6
        RFT     777777
        TXI     MASTER,7,-3             FINAL EXIT, NO MORE PLANES TO EXAMINE
        PIA
```

```
        PDX     ,5
        SXA     *+2,5
        XCL                             PUT BMAST INTO ACC
        LDI     **,4                    ** IS BASE ADDRESS OF PLANE
        ONT     COMMON                  COMMON HAS XMASK
        TXI     *+3,7,-1                JUMP IF POINT P=0
READB   ORS     **                      **=ZWORD, SET BIT IN ZWORD=1
        TXI     *+4,7,-1
        LDI     **                      **=ZWORD
        RIA                             SET BIT=0 IN ZWORD (IN SR)
        STI     **                      **=ZWORD
        XCL                             RETURN BMASK TO MQ
        RQL     35                      ADVANCE BMASK
        TXI     READA,6,-1
*
*
*
*       WRITE - BLACKENS A SELECTED POINT IN A SELECTED PLANE
*
WRITE   AXT     2,1
        TSX     MORDER,4
        CLA     =1
        LDI     =014
        TSX     MODIFY,4
        CLA     =144
        LDI     =01
        TSX     MODIFY,4
        LXD     ORDERM,4
        SXA     *+2,4                   SET PLANE BASE ADDRESS INTO *+2
        TSX     XY,3                    SET XR4,ACC, ACCORDING TO COORDINATES
        ORS     **,4                    **=PLANE,  WRITE POINT (X,Y)
        TXI     MASTER,7,-2
*
*
*
*       WRITEP - READ COORDINATES FROM LOCATION NAME, THEN BLACKEN POINT.
*
WRITEP  AXT     2,1
        TSX     MORDER,4
        CLA     =1
        LDI     =010
        TSX     MODIFY,4
        LAC     ORDERM+1,4
        CLA     ,4
        STO     ORDERM+1
        TRA     WRITE+5
*
*
*
*       XY      GIVEN X,Y COORDINATES IN LOCATION ORDERM+1, THIS ROUTINE
*               EXITS WITH 'MASK' IN ACCUMULATOR, XR1 DESTROYED, XR4 SET
*               SO THAT 'MASK' ON PLANE,4 SELECTS DESIRED POINT OF PLANE
*               ENTER VIA  TSX XY,3       RETURN IS TO 1,3
*
XY      CLA     EXMODE
        SUB     =72
        LXA     ORDERM+1,4              XR4=Y
```

```
        TZE     READ7                   TRANSFER IF IN 72-MODE
        PXA     ,4
        SUB     =35                     ACC ADDRESS HAS 35-Y
READC   PAC     ,4                      XR4=(M-Y) COMPLEMENTED
        LDC     ORDERM+1,1              XR1=X (COMPLEMENTED)
        CAL     =C4C0000000000
        ARS     ,1                      ACC HAS XMASK
        TRA     1,3
READ7   PXA     ,4
        SUB     =71                     ACC HAS 71-Y
        LXD     ORDERM+1,1              XR1=X
        TXL     READC,1,35              JUMP IF POINT P IN LEFT-HALF-PLANE
        SUB     =72                     ACC ADDRESS IS 143-Y
        PAC     ,4
        LDC     ORDERM+1,1
        TXI     READC+2,1,36            XR1 HAS (X-36) COMPLEMENTED
*
*
*
*       MARK - S1,S2,D,S3 - P IN S1 IFF P HAS A -D-NGHBR IN S2, P IN S3
*
MARK    AXT     4,1
        TSX     MORDER,4
        CLA     =144
        LDI     =0105
        TSX     MODIFY,4
        CLA     =2
        LDI     =010
        TSX     MODIFY,4
        CLA     ORDERM
        PDX     ,1
        SXA     MARK7+1,1
        CLA     ORDERM+1
        STD     MARK1+1
        STA     MARK2+1
        CLA     ORDERM+3
        PDX     ,1                      XR1=S3
        TXL     *+4,1,0                 JUMP IF S3 IS NOT SPECIFIED
        STD     MARK6
        CLA     6MARK
        TRA     *+2
        CLA     7MARK
        STD     MARK2+2
        TXI     MASTER,7,-2
6MARK           ,,MARK6
7MARK           ,,MARK7
        BCOLOP  MACH1,,CLEAR
MARK1   BCOLOP  MACH2,,EQUAL,**         **=S2
MARK2   SETSET  **,,MARK3,COMMON,**     **=DL, MARK6/MARK7
MARK3   SETTRA  **,MARK4                **=+D, BY MARK2
MARK4   SHIFT   MACH2,,**,**            **,**=-D BY MARK 3
        BCOLOP  MACH1,,OR,MACH1,,MACH2
MARK5   JUMP    UC,MARK1
MARK6   BCOLOP  MACH1,,AND,MACH1,,**        **=S3
MARK7   BCOLOP  **,,EQUAL,MACH1         **=S1
        JUMP    RETURN,,25
*
```

```
*
*
*       CONNEC - PUT IN S1 THE GROWTH OF POINTS OF S2 IN DL-DIRECTIONS
*
CONNEC  AXT     4,1
        TSX     MCRDER,4
        CLA     =144
        LDI     =0105
        TSX     MODIFY,4
        CLA     =2
        LDI     =010
        TSX     MODIFY,4
        CLA     ORDERM
        PDX     ,1                      XR1=S1
        SXD     CHAIN2,1
        SXD     CHAIN3,1
        SXD     CHAIN6+1,1
        SXA     CHAIN5+1,1
        CLA     ORDERM+1
        STD     CHAIN1+1                STORE S2
        STA     CHAIN2+1                STORE DL
        CLA     ORDERM+3
        STD     CHAIN2+3                STORE S3
        STD     CHAIN1
        TXI     MASTER,7,-2
        INDEX   SET,21,-1
CHAIN1  BCOLOP  MACH3,,AND,**,,**       **=S2,S3
CHAIN2  MARK    **,,**,,MACH3,,**       **=DL,S1,S3
CHAIN6  BCOLOP  MACH4,,AND,**,,MACH3,,BAR       ** IS S1
CHAIN3  BCOLOP  MACH3,,OR,MACH3,,** **=S1
        INDEX   INCR,21,1
CHAIN4  JUMP    NONULL,CHAIN2,MACH4
CHAIN5  BCOLOP  **,,EQUAL,MACH3         **=S1
        JUMP    RETURN,,24
*
*
*
*       TCHAIN - PUT P IN S1 IFF THE POINT P IN S2 IS A BEGINNING POINT
*                FOR A CHAIN IN S3 WITH A LENTH L SATISFYING SOME RELATION
*
TCHAIN  AXT     3,1
        TSX     MCRDER,4
        CLA     =144
        LDI     =015
        TSX     MODIFY,4
        CLA     =2
        LDI     =040
        TSX     MODIFY,4
        CLA     =1
        LDI     =020
        TSX     MODIFY,4
        CLA     ORDERM
        STD     TCH1
        STD     TCH2+3
        CLA     ORDERM+1
        STD     TCH1+1                  SET S2
        STA     TCH8+1                  SET S1
```

```
        CLA     ORDERM+2
        STD     TCH3+1
        PAC     ,5                      XR5=-DL
        STA     TCH2+1                  SET DL
        CAL     1,5                     ACC HAD DIRECTION LIST
        ARS     32                      ACC=D
        STA     TCH4+1                  SET D
        STA     TCH6+1                  SET D
        LDI     3,7
        RNT     3
        TXI     *+4,7,-2
        CLA     TCH1+2                          SET EQUAL
        TXI     *+3,7,-2
                ANDCOM
        CLA     *-1
        STA     TCH8                    SET ANDCOM/EQUAL
        RFT     20
        TRA     *+5                     JUMP TO SET S2=BN
        RNT     10
        TRA     *+5                     JUMP TO SET S2=B1
        CLA     TCH2                            SET S2=B(N+1)
        TRA     *+4
        CLA     TCH2+1
        TRA     *+2
        CLA     TCH1+2
        STD     TCH8+1                  SET S2=B1/BN/B(N+1)
        RFT     70
        TRA     *+3                     JUMP TO SET S3=B(N+1)
        CLA     TCH1+3                          SET S3=BN
        TRA     *+2
        CLA     TCH2
        STD     TCH8
        PXA     ,7
        STA     IR-1+24                 FAKE LINK INTO IR24
        CLA     TCH3+2
        TRA     JUMPUC+1                FAKE JUMP TO TCH1
TCH1    BCOLOP  MACH4,,AND,**,,**       MACH4=B1, **=S2,S3
        BCOLOP  MACH6,,EQUAL,MACH4,,MACH6
        INDEX   SET,22,1
TCH2    MARK    **,,MACH5,,MACH4,,**        MACH5=E(I+1), MACH4=EI,
TCH3    JUMP    EXACT,TCH4,22,**        **=N                    **=DL,S3
        BCOLOP  MACH4,,EQUAL,MACH5,,TCH1
        JUMP    NULL,TCH8,MACH5
        INDEX   INCR,22,1
        JUMP    UC,TCH2
TCH4    SETTRC  **,TCH5
        SETTRB  22,TCH5
TCH5    SHIFT   MACH5,,**,**
TCH6    SETTRC  **,TCH7                 **=D
        INDEX   DECR,22,1
        SETTRB  22,TCH7
TCH7    SHIFT   MACH4
TCH8    BCOLOP  **,,**,**,,**,,**       **=S1,ANDCOM/EQUAL, ETC. ETC.
        JUMP    RETURN,,24
*
*
*
```

```
*       TMARK - FUNCTIONS AS APPLY TEMPLATE AND BUBBLE
*
TMARK   AXT     4,1
        TSX     MORDER,4
        CLA     =144
        LDI     =015
        TSX     MODIFY,4
        CLA     =2
        LDI     =040
        TSX     MODIFY,4
        CLA     =1
        LDI     =02C
        TSX     MODIFY,4
        CLA     IR                      SAVE IR1, IR3, IR4
        STO     IR-1+21
        CLA     IR-1+3
        STO     IR-1+23
        CLA     IR-1+4
        STO     IR-1+24
        CLA     ORDERM
        STD     TM3+1                   SET S3
        CLA     ORDERM+1
        STD     TM9+1                   SET S2
        STA     TM8+1                   SET S1
        CLA     ORDERM+2
        STA     TM2+1                   SET DL
        PDX     ,1                      XR1=N
        LDI     3,7
        PXA     ,1                      ACC=N
        STT     ORDERM+2
        CAS*    ORDERM+2                COMPARE N TO NO OF DIRECTIONS IN LIST
        TXI     TM11,7,-2               NOGO, CHEAT FOR QUICK ANSWER
        TXI     TM10,7,-2               OK UNLESS REL=GT
TM12    RNT     42                      SKIP IF LE OR GT  (N BECOMES N+1)
        TXI     *+1,1,-1
        TXI     *+1,1,1
        PXA     ,1
        RNT     1                       SKIP IF REL=EQ,GE,GT
        TXI     *+3,7,-2
        STO     IR-1+3
        TXI     *+4,7,-2
        STO     IR-1+4
        STZ     IR-1+3
        TRA     FKL25
        RNT     3                       SKIP IF REL=GE,GT
        TXI     *+3,1,1
        CLA     =10
        TRA     *+2
        PXA     ,1
        STO     IR-1+4
FKL25   SXA     IR-1+25,7               FAKE LINK INTO IR25
        CLA     TM9
        TRA     JUMPUC+1                FAKE JUMP TO TM1
TM1     INDEX   SET,1,0
TM9     BOOLOP  MACHO,,EQUAL,**,,TM1        **=S2
TM2     SETSET  **,,TM4,COMMON,TM7          **=DL
        INDEX   INCR,1,1
```

```
TM3     BOOLOP  MACH0,1,EQUAL,**              **=S3
TM4     SETTRC  **,TM5                       **=D BY TM2
TM5     SHIFT   MACH0,1,**,**                **=X,Y BY TM4
        BOOLOP  MACH0,1,AND,MACH0,1,MACH MACH0,1 HAS POINTS OF S2 WHICH
        JUMP    UC,TM2                       HAVE D-NEIGHBORS IN S3
TM7     BUBBLE  ,1,MACH1,,-1,1
        BOOLOP  MACH1,1,CLEAR
        BOOLOP  MACHT,,CLEAR
TM8     BOOLOP  **,,AND,MACH0,3,MACH0,4,BAR      **=S1
        INDEX   TRANS,1,21
        INDEX   TRANS,3,23
        INDEX   TRANS,4,24
        JUMP    RETURN,,25
TM10    RNT     53
        TXI     TM12,7,2            REL IS NOT GT, CONTINUE NORMALLY
        AXT     ,2                  FAKE CHANGE OF USER'S TMARK ORDER
        TRA     SETNUL+6              INTO A  'BOOLOP  S1,,CLEAR'  ORDER
TM11    RFT     1                   SKIP IF REL=LT OR LE
        TRA     *-3                 REL=EQ,GT, OR GE, GET QUICK ANSWER
        AXT     -1,2                FAKE CHANGE OF USER'S TMARK ORDER
        TRA     SETTRU+6              INTO  'BOOLOP  S1,,EQUAL,S2'  ORDER
*
*
*
*       BOOFUN - COMPUTES, FOR POINTS OF S2, A BOOLEAN FUNCTION OF THE
*       EIGHT NEIBORING POINTS, USING VALUES FROM S3.  ANSWER TO S1.
*
BOOFUN  AXT     2,1
        TSX     MORDER,4
        CLA     =144
        LDI     =C15
        TSX     MODIFY,4
        SXA     IR-1+22,7           SAVE XR7
        CLA     IR-1+7              SAVE IR7
        STO     IR-1+25
        STZ     IR-1+7
        CLA     ORDERM
        STD     GD88+1
        CLA     ORDERM+1
        STA     EV7+1
        STD     EV7
        ZAC
        PAI
GD1     AXT     6,6
        LDQ     2,7
GD2     ZAC
        LGL     6
        CAS     =C53
        TRA     *+2
        TRA     GD4                     $ HAS BEEN READ
        CAS     =8
        TRA     GD3                 IGNORE CHARACTER
        NOP
        PAC     ,1
        XEC     *+2,1
        TRA     GD3
        SIR     1
```

```
        SIR     2
        SIR     4
        SIR     10
        SIR     20
        SIR     40
        SIR     100
        SIR     200
        SIR     400
GD3     TIX     GD2,6,1
        TXI     GD1,7,-1
GD4     AXT     ,1                      XR1 WILL HAVE NO. OR DIRECTNS IN LIST
        LDQ     =1                      CLEAR MQ
        AXT     ,2                      XR2 WILL RUN THRU ALL POSS. DIRECTNS
GD5     XEC     B1OR0+35,2              IS D=V(XR2) INCLUDED$
        TXI     BFDL+2,1,1                YES
        TXI     *+1,2,1                   NO
        TXL     GD5,2,8
        SXA     BFDL,1
        STQ     BFDL+1
        AXC     *+2,7
        TRA*    *+1
GD6     SETSET  BFDL,,GD7,COMMON,GD9
GD7     SETTRC  **,GD8
GD88    BCOLOP  MACHT,,EQUAL,**             **=S3
GD8     SHIFT   MACHT
        J7094M  3
        CLA     GD7+1
        STA     IR-1+7                  SET IR7=D
        TRA     MASTER
GDC     BCOLOP  MACH0,7,EQUAL,MACHT
        JUMP    UC,GD6
GD9     BCOLOP  MACHT,,CLEAR
        BCOLOP  MACH9,,CLEAR,,,,,BAR
        J7094C  EVBFUN
BFDL    PZE
        PZE
        PXA     ,2
        LGR     4
        TRA     GD5+2
EVBFUN  LXA     IR-1+22,1               SET XR1=ORIGINAL XR7
        AXT     ,3
EV1     AXT     6,6
        LDQ     2,1
EV2     ZAC
        LGL     6
        CAS     =053
        TRA     *+2
        TRA     EV6                         $ HAS BEEN READ
        CAS     =061
        TRA     *+2
        TXI     EV4,3,2                 / HAS BEEN READ
        CAS     =020
        TRA     *+2
        TRA     EV5                     + HAS BEEN READ
        CAS     =8
        TRA     EV4                     IGNORE CHARACTER
        NOP
```

```
        STA     IR-1+7                  V(IR7)=D
        TSX     SAVE,4
        AXC     *+2,7
        TRA     ANDTRU,3
        BOOLOP  MACH9,,,MACH9,,MACH0,7
        J7094C  EV3
EV3     TSX     RESTOR,4
        AXT     ,3
EV4     TIX     EV2,6,1                 ENTER HERE TO READ NEXT CHARACTER
        TXI     EV1,1,-1
EV5     TSX     SAVE,4
        AXC     *+2,7
        TRA     ORTRUE,3
        BOOLOP  MACHT,,,MACHT,,MACH9
        BOOLOP  MACH9,,CLEAR,,,,,BAR
        J7094C  EV3
EV6     SXA     IR-1+23,1
        AXC     *+2,7
        TRA     ORTRUE,3
        BOOLOP  MACHT,,,MACHT,,MACH9
EV7     BOOLOP  **,,AND,MACHT,,**       **=S1,S2
        INDEX   TRANS,7,25
        J7094C  EV8
EV8     LXA     IR-1+23,7
        TXI     MASTER,7,-3
SAVE    SXA     IR-1+24,6
        SXA     IR-1+23,1
        STQ     IR-1+21
        TRA     1,4
RESTOR  LXA     IR-1+24,6
        LXA     IR-1+23,1
        LDQ     IR-1+21
        TRA     1,4
*
*
*
*       SYMBOL DEFINITIONS AND BLOCK RESERVATIONS
*
COMMON  BSS     150
IR      BSS     25                      BLK RESERVE SPACE FOR 25 INDEX RGTRS
IWW     BSS     200
ORDERM  BSS     21
ORDERS  BSS     14600
PATLBL  BSS     14
PLANES  BSS     144*51
TPLANE  BSS     144*11
MACH0   EQU     0*144+TPLANE
MACH1   EQU     1*144+TPLANE
MACH2   EQU     2*144+TPLANE
MACH3   EQU     3*144+TPLANE
MACH4   EQU     4*144+TPLANE
MACH5   EQU     5*144+TPLANE
MACH6   EQU     6*144+TPLANE
MACH7   EQU     7*144+TPLANE
MACH8   EQU     8*144+TPLANE
MACH9   EQU     9*144+TPLANE
MACHT   EQU     1440+TPLANE
```

```
M0      EQU     PLANES
M1      EQU     01*144+PLANES
M2      EQU     02*144+PLANES
M3      EQU     03*144+PLANES
M4      EQU     04*144+PLANES
M5      EQU     05*144+PLANES
M6      EQU     06*144+PLANES
M7      EQU     07*144+PLANES
M8      EQU     08*144+PLANES
M9      EQU     09*144+PLANES
M10     EQU     10*144+PLANES
M11     EQU     11*144+PLANES
M12     EQU     12*144+PLANES
M13     EQU     13*144+PLANES
M14     EQU     14*144+PLANES
M15     EQU     15*144+PLANES
M16     EQU     16*144+PLANES
M17     EQU     17*144+PLANES
M18     EQU     18*144+PLANES
M19     EQU     19*144+PLANES
M20     EQU     20*144+PLANES
M21     EQU     21*144+PLANES
M22     EQU     22*144+PLANES
M23     EQU     23*144+PLANES
M24     EQU     24*144+PLANES
M25     EQU     25*144+PLANES
M26     EQU     26*144+PLANES
M27     EQU     27*144+PLANES
M28     EQU     28*144+PLANES
M29     EQU     29*144+PLANES
M30     EQU     30*144+PLANES
M31     EQU     31*144+PLANES
M32     EQU     32*144+PLANES
M33     EQU     33*144+PLANES
M34     EQU     34*144+PLANES
M35     EQU     35*144+PLANES
M36     EQU     36*144+PLANES
M37     EQU     37*144+PLANES
M38     EQU     38*144+PLANES
M39     EQU     39*144+PLANES
M40     EQU     40*144+PLANES
M41     EQU     41*144+PLANES
M42     EQU     42*144+PLANES
M43     EQU     43*144+PLANES
M44     EQU     44*144+PLANES
M45     EQU     45*144+PLANES
M46     EQU     46*144+PLANES
M47     EQU     47*144+PLANES
M48     EQU     48*144+PLANES
M49     EQU     49*144+PLANES
M50     EQU     50*144+PLANES
SET     EQU     SETIR
INCR    EQU     INCREM
DECR    EQU     DECREM
BAR     EQU     2
CLEAR   EQU     SETNUL
EQUAL   EQU     SETTRU
```

```
AND     EQU     ANDTRU
OR      EQU     ORTRUE
EXOR    EQU     XORTRU
UC      EQU     JUMPUC
NULL    EQU     JUMPNL
NONULL  EQU     JUMPNN
ZERO    EQU     JUMPZE
NOZERO  EQU     JUMPNZ
LESS    EQU     JUMPLT
EXACT   EQU     JUMPET
MORE    EQU     JUMPGT
LINK    EQU     JUMP
RETURN  EQU     JUMPRS
DUMP    EQU     1
LT     8EQU     02
LE     8EQU     42
EQ     8EQU     71
GE     8EQU     23
GT     8EQU     53
*
*
*
        END     MASTER-2
```

The following scheme may be used to incorporate a trace routine into PAX for the purpose of code checking PAU programs. This addition to PAX causes the location of every PAU instruction to be printed just before it is executed.

The following SCAT "program" is placed at the end of the Order Deck (see Appendix I.1, Report No. 147):

```
TRACE    PXA     ,7
         PAC     ,7
         TXL     TRACF,7,ØRDERS-1
         TXH     TRACF,7,MO-15
* THE ABOVE TWØ ØRDERS TRANSFER IF THE PAX INSTRUCTION
* IS ØUTSIDE THE REGION RESERVED FOR USER'S PAX ØRDERS
         SXA     TRACG,7
         CALL    .PRINT
         STR     TRACG+1
         STR     TRACG
         STR
TRACF    PXA     ,7
         PAC     ,7
         TRA*    ,7
TRACG    PZE
         BCI     *,1H ,Ø7*
         ØRG     PAX
         TRA     TRACE
```

The above program refers to these locations within the PAX program symbolically, i.e., ØRDERS-1, PAX, and MO-15.

The test which determines whether the value of XR7 lies within the range ØRDERS $\leq$ V(XR7) $\leq$ MO-14 is necessary because execution of macro orders (see Section 5) causes PAX to execute internal coding which lies outside this region.

The orders SETTRA, SETTRC, SETTRB, and SETSET described in Section 7 may be used in PAX provided the user defines them by placing the following macro definitions in the Order Deck prior to the first occurrence of the orders therein:

```
SETTRA    MACRØ   D,NAME
          VFD     36/DlALL+2,18/NAME,18/D
          END
SETTRC    MACRØ   D,NAME
          VFD     36/DLALL+41,18/NAME,18/D
          END
SETTRB    MACRØ   K,NAME
          VFD     36/DLALL+22,18/K,18/NAME
SETSET    MACRØ   DL,K,NAME1,NAME2,NAME3
          VFD     18/NAME1,18/DLALL+49,18/NAME2,3/K,15/DL
          VFD     18/NAME3,18/JUMP-31,36/0
          END
```

APPENDIX V

STRUCTURE OF THE BINARY PAX DECK

```
$       SCATRE
$       GO
$       DUMP
        UNLIST
        ENDPGM
        BINARY
(Binary Deck containing symbol definitions and program)
(Macro definitions, as given in Appendix I)
        NØCRS
        PMC
        ØRG     ØRDERS
        LIST
        EJECT
(Order Deck goes here)
 620708END
```